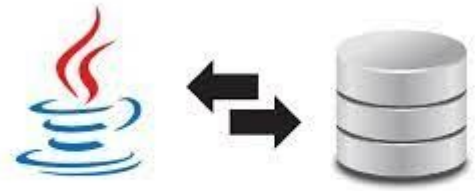
FACULTAD DE INGENIERIA
UNIVERSIDAD NACIONAL DE JUJUY

Introducción a JPA

Programación Orientada a Objetos

San Salvador de Jujuy

UNJu – Facultad de Ingeniería



Ing. José Zapana

Java Persistence API

- Java Persistence API es la API de persistencia desarrollada para la plataforma Java EE
- Es un framework del lenguaje de programación Java que maneja datos relacionales en aplicaciones usando la Plataforma Java en sus ediciones Standard (Java SE) y Enterprise (Java EE).

Entidades

- Una entidad de persistencia (**entity**) es una clase de Java ligera, cuyo estado es persistido de manera asociada a una tabla en una base de datos relacional.
- Los metadatos del objeto/relacional pueden ser especificados directamente en la clase, usando las anotaciones de Java (annotations), o en un documento descriptivo XML, el cual es distribuido junto con la aplicación.
- Características de una Entity
  - Proporcionar un constructor por defecto (ya sea de forma implícita o explícita)
  - Ser una clase de primer nivel (no interna)
  - No ser final
  - Implementar la interface java.io.Serializabe si va a ser accedida remotamente

Entity - Anotaciones

- @Entity (name = “”)
- @Id
- @Transient
- @GeneratedValue
- @Column
- @Temporal (TemporalType.DATE / TemporalType.TIMESTAMP)
- Otros…

Gestión de una Entity

- Se utiliza el patrón DAO
- CrudRepository del cual se herada la interface DAO
- Con lo cual Spring Boot inyecta una implementación de ese DAO
- Solo es necesario agregar los método específicos de esta entidad

```
@Repository
public interface PlayerRepository extends CrudRepository<Player, Long> {

    List<Player> findByTeamId(long teamId);
}
```

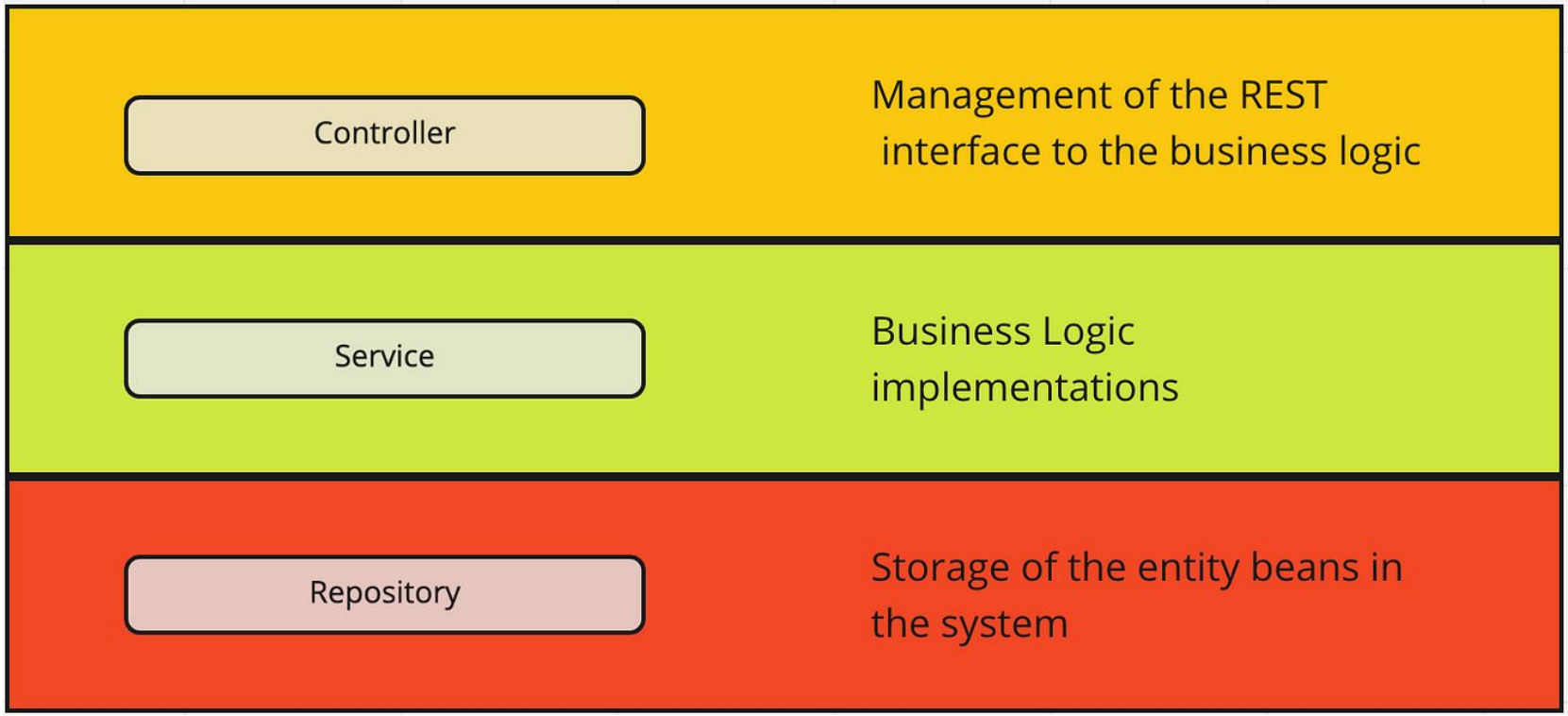
CrudRepository

Inyecta las siguientes operaciones

- count(): long
- delete(id: ID): void
- delete(entity: T): void
- delete(entities: Iterable<?extends T>): void
- deleteAll(): void
- findOne(id: ID): T
- findAll(ids: iterable<ID>): Iterable<T>
- findAll(): Iterable<T>
- save(entity: T): T
- save(entities: iterable<T>): Itereable<T>

Capas de una aplicación

Capas básicas para diseñar una aplicación con Spring Boot



Servicios

- Contienen la lógica de negocio de la aplicación
- Son el nexo entre los controladores y los repositorios DAO
- Sus métodos utilizan a los **DAOs** para realizar las operaciones con la Base de Datos

Analice el ejemplo citado en el link de la última diapositiva

Conexión con la base de datos

- application.properties (ejemplo)

```
server.port=9000
spring.datasource.url=jdbc:mysql://localhost:3306/pv2020-tienda?serverTimezone=GMT-3
spring.datasource.username=root
spring.datasource.password=

# Configuración complementaria MySql
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MariaDBDialect
spring.jpa.hibernate.ddl-auto = update
spring.jpa.properties.hibernate.jdbc.lob.non_contextual_creation=true
spring.jpa.show-sql=true
spring.jpa.hibernate.dll-auto=update
```

Ejemplo práctico

https://dzone.com/articles/spring-boot-jpa-hibernate-oracle